Interactive management of time series

Carlo Bianchi and Giorgio Calzolari and Paolo Corsi

IBM Scientific Center, Pisa, Italy

1974

INTERACTIVE MANAGEMENT OF TIME SERIES

Carlo BIANCHI - IBM Italy, Pisa Scientific Center

Giorgio CALZOLARI - IBM Italy, Pisa Scientific Center

Paolo CORSI - IBM Italy, Pisa Scientific Center

May 1974

ABSTRACT

At the Pisa Scientific Center an interactive package has been developed under CP-67/CMS, which is particularly helpful when the data to be processed are time series.

The interactive facilities of CP-67/CMS [1, 2] are strenghtened in such a way as to allow an easy interactive correction procedure during the execution of any command.

The central file of time series data is available to be interactively shared among several users. Each user can also keep and use his own private series.

Time series can have annual, semestral, quarterly or monthly entries. Arithmetic, algebraic and trigonometric operators, special operators — mean, variance, maximum, minimum, lagging, variation rate, first difference and selection of a part of a series — are supplied for interactive use. When applied to a time series, monadic operators — logarithm, exponential, etc — return a new series with different numeric data. When applied to two time series with the same periodicity (e.g. both with annual data, etc) dyadic operators — sum, subtraction, etc — return a new series referring to the period common to both the original series.

Three interactive methods of estimation are supplied: ordinary least squares (OLS), two stage least squares (TSLS), limited information single equation (LISE). Correlation matrix and graphic plotting routines are provided.

In addition it is easy for users with some experience of computer processing to insert special functions in the IMTS environment.

CONTENTS

1. INTRODUCTION

A significant element in studying the structure of a national economy is the organization of a data file consisting of the observations, over time, on the variables which characterize the economy.

The main purpose of this data base is to offer researcher a supply of information which can be increased, deleted or updated during the research.

Whenever data must be manipulated in a repetitive way, the application of some form of computer processing can be of great help.

These fundamental needs are much better answered by an interactive approach which, by allowing a direct dialogue between the user and the computer, facilitates this iterative procedure.

The interactive language developed at the Pisa Scientific Center of IBM Italy makes it possible for a large number of users to have simultaneous access to the same information, without interfering with one another.

Results of each computation are printed on the user terminal. Therefore, he can rapidly execute complex researches in which each step depends on the results of preceding steps.

In order to avoid any conflict resulting from simultaneous updatings, only one virtual machine can mantain the time series file shared among the users.

Data to be stored may be prepared by any user: he can use them in his own disk for particular purposes or request their inclusion into the general central file.

The central file holds various series of data which are handled globally by one single program, called IMTS (Interactive Management of Time Series).

By means of this program it is possible to use data both from the central file and from a private single user file; these data are accessed using only the abbreviated name of the series to be used. The user has only to know the abbreviated names of the series, whose complete name may be obtained by issuing a command at the terminal.

Summarizing, the research capabilities supplied by te IMTS package are analysis, transformation and regression of stored data.

These features can be used independently or in any combination by means of the IMTS language, which provides a single command for each application: it is also possible to write regression equations as well as algebraic expressions defining transformations of the econometric variables.

Simulation by econometric models is already performed at the Pisa Scientific Center by means of the programming system DMS/2 [3]. An interface between the two packages is obtained by means of the command DMS (see chapter 4), issued in the IMTS environment.

As regards analysis and transformation of stored data, on the time series data a lot of operators, frequently used during the economic analysis, can be applied.

They are described in detail in the chapter 4.

With regards to the estimation of the parameters of a model, three methods are available at present: Ordinary Least Squares (OLS), Two Stage Least Square (TSLS) and Limited Information Single Equation (LISE); they are also described in chapter 4.

Any user, with some experiences of Fortran and/or Assembler languages, can create special functions (additional estimation methods, particular cross-sections studies, etc) by means of subroutines which, after having been compiled, can be used immediately in the IMTS environment, in conjunction with any other standard or user's function.

For the standard commands, however, no programming skill is required; the researcher can devote all his attention to problem analysis. The research capabilities are supported by many other capabilities, such as the on-line formal errors recovery procedure (see chapter 4) or the Macro facility (see chapter 3).

Taking into consideration the experimental character of the language, it is fully understandable that certain functions are incomplete or missing: the improvement and the completion of the language is expected to be one of the main results of the IMTS usage.

At present, new functions are frequently inserted in the IMTS package, according to the necessity of the user in the research area.

The system was originally designed for use in econometrics, but it is expected to be applicable to many other fields such as management science or social science.

2. MAIN STORAGE MANAGEMENT

The storage may be divided into four areas:

1) Core of the CMS operating system, free area for system functions and transient area (0-12000 hex.).

2) Low address free area, designed to hold the time series work matrix, or to load the system's or user's module format programs (EDIT, COMBINE, SORT, NEW, CREARCH, APLIUP, etc.)
(12000-36000 hex.).

3) Protected area, containing the program routines permanently held in the storage (static routines), such as the coding and decoding routines, I/O routines, etc.
(36000-50000 hex. approx.).

4) High address free area, intended for the dynamic loading of plotter or regression routines (50000 - end of virtual memory).

In order to maintain that subdivision, before interpreting each command, the program operates on th NUCON table of the CMS, changing 5 addresses as follows:

LOWEXT (lower end of the high-storage addresses occupied by the operating system via FREE and EXTEND routines): the value is removed and replaced by 34000 (hex.); whatever module may be executed (of the system or of an user), or TEXT module loaded, no storage area may be allocated (eg: used) above the specified address.

LOCCNT (location counter) and HIMAIN (upper end of the low-storage addresses allocated by the GETMAIN routine of the operating system): the content is removed and replaced by 12000 (hex.) in both cases.

LSTADR (address of the last page of the virtual machine memory) and LDRTBL (last address in the loader tables, practically speaking the same as the last address in the virtual memory): the content is removed and replaced by 35000 and 36000 (hex.) respectively.

Immediately afterwards, the program checks whether or not the command to be executed is a CMS command. If so, the CMS takes over and executes the command, and then returns to the 'calling' program, which resets the addresses changed in the NUCON table.

If not, the addresses changed in NUCON are immediately reset and control is passed over to the coding and decoding routines (permanently housed in the storage). These routines use the free area (12000-36000) organized as the work matrix.

If the command given calls for the use of a regression or plotter routine not resident in the storage, the requested routine is dynamically loaded by the CMS loader into the high-address free area and, among other parameters, the address of the work matrix is passed to it.

Any CMS command may be executed. However, it is not possible to execute chained commands relating to a common storage area (for example, LOAD and START for a program, though LOAD (XEQ) is possible and so is $).

In particular, the following CMS commande are frequently needed:

EDIT
ALTER
COMBINE
LISTF
STAT
PRINT and OFFLINE PRINT
SORT
APLIUP

to which are also added:

NEW
CREARCH

which are the names of utility programs (contained in module format in the virtual machine which contains the time series file) loaded and executed by the CMS in the same way as the modules which may be handled by the system.

3. FILE MANAGEMENT

The input/output routines of the program provide for the maintenance of four classes of files.

The first is made up of the permanent time series file available to all users.

The second is also made up of only one file, containing the abbreviated names of all the permanent time series. It is the directory.

The third class of files is made up of all the files containing the private time series of individual users, or the time series obtained as intermediate results of one procedure, for use in subsequent ones.

Last, the fourth class includes only one file in which the commands to be executed are stored as and when they arise. It is used both for correcting formal errors in commands, and for carrying out burdensome sequences of command, which may be stored once and for all and then recalled as and when needed.

The I/O routines require fixed length records file.

3.1 Time series file

CMS Identifier: ARCHIVIO SERIESTO P1

This is a direct access file made up of 2100 records with a fixed length of 800 bytes.

Each record designed to contain one single time series holds the following information:

1) The complete name of the time series
2) The abbreviated name of the series
3) The source of the data
4) The unit of measurement
5) A code to indicate whether the data are annual, semi-annual, quarterly or monthly
6) The first year
7) The last year
8) The first quarter (or half-year or month)
9) The last quarter (or half-year or month)
10) The number of data in the series
11) The numerical data
12) A code indicating whether the data are continued on another record (this occurs only in the case of monthly data covering at least 14 years).

The data must be complete from beginning to end (missing data are not allowed).

The maximum number of items in a series is 360.

The serial number of records to be read (or written) is passed (as a parameter) to the routine which handles file input/output.

Records, without format, are read and allocated to an appropriate 800 bytes buffer (or read from it).

3.2 Directory

CMS Identifier: TABELLA ABBREV P1

This file contains 210 records with a fixed length of 80 bytes (card format). Each time series is distinguished by its own abbreviated name.

Each directory record contains 10 abbreviated names of time series (8 bytes each). The name contained in record x, position y, corresponds one-to-one to the record $10(x-1)+y$ of the file.

The last 100 records of the file and the last 10 in the directory are reserved for long time series, that must continue in a new record.

There cannot be more than one series with the same abbreviated name, but all the other data may be common to various series.

The file is read at the beginning of the work session and allocated to an appropriate tabel held in the (static) storage. The procedure for looking up a series is as follows: the program scans the table (in the storage), comparing the abbreviated names of the time series contained therein with the name requested. After finding the name in question, the program obtains the serial number and passes this to the I/O routine which finally reads from the file the record containing the requested series. The record contains a flag which indicates whether or not the series is continued in another record; when the series is continued, the program scans the last positions on the table in search of the identifier 'nnnn****', where nnnn is the serial number of the series, and proceeds as described above.

The numerical values of the time series are allocated to a column on the work matrix.

3.3 Work files and private time series

CMS Identifier: FILE (abbreviated-name) P1

As well as the central file, users may also use the private time series contained in their own work disk. In such cases, each time series is a CMS file, and the relative FILETYPE coincides with the abbreviated name of the series in question.

In all such files, the records must be of fixed length: 80 bytes (card format), with the following layout:

First card

col.	1 - 4	Serial number of the series (unused if the series is not to be stored into the central file)
	5 - 12	Abbreviated name
	13	Division into periods (Y = annual data) (S = semi-annual data) (Q = quarterly data) (M = monthly data)
	14 - 15	First year (last two digits)
	16 - 17	First period (eg: 01 for monthly data is referred to January, for semi-annual data is referred to the firs half of the year; for annual data 01 is mandatory).
	18 - 19	Last year
	20 - 21	Last period
	22 - 25	Number of data
	26 - 75	Complete name of the series
	76 - 77	Initial to indicate what is involved: index numbers, levels, first differences, or variation rates (IN, LE, FD, RV respectively).

Second card

col.	1 - 32	Data source
	33 - 52	Unit of measurement

From third card onwards

col.	8 - 22	Values of the series in sequence (format F15.6 or E15.8)

Last card (end of file)

col.	1 - 7	ENDFILE

If data are prepared on cards, the above formats must be respected. However, to input new series from the terminal, or to update those already existing, it is possible to use a number of programs (as described hereunder) which relieve the user of having to keep to these formats.

The work files, obtained from time series contained in files or in the central file after procedures of various kinds (e.g.: the addition of three series), have the same format, except for the positions relating to the complete name, data source and unit of measurement, which are left blank, as they tend to lose significance. Should the user decide to retain these files, attributing a particular significance to them, they may be updated by means of the EDIT program of the CMS.

In order to write a new series into the central file, it is necessary to prepare a file of this kind and the CMS identifier must be FILE FT01F001.

It is possible to house more than one series one after the other in the same file. The update program (CREARCH) is started by the virtual machine wich has the responsability of file maintenance.

The update program reads the file and carries out the following checks:

1) The actual number of data must coincide with that indicated.

2) The number of data must coincide with that obtainable from the first and last year and from the number of data for each year. In the event of a negative result for either of these checks, the update operation is not executed.

3) No series with the same abbreviated name should be held in the file. If not so, the program requests authorization to replace it with the new series.

4) The record corresponding to the given number indicated should be vacant. If it is not vacant, the update is nevertheless executed and the series is inserted in the first vacant record, signalling the relevant number to the terminal.

5) If the series needs a continuation record, the programs seeks the first vacant position in the space reserved for this purpose. The corresponding directory space is indicated 'nnnn****', nnnn being the number of the record containing the first part of the series.

3.4 File for the correction of formal errors

CMS Identifier: GO IMTS P1

The commands given during a work session at the terminal may contain formal errors (unclosed brackets or quote marks, illegal symbols, etc.). In the case of a complex command in particular, it is convenient to allow the user to correct the given command, instead of entirely rewriting it. All instructions which pass through the coding and decoding routines (CMS commands are thus excluded) are stored in the GO IMTS file, starting from the first record.

The file contains records of a fixed length of 130, that is the same length as an input line at the typewriter terminal. In the event of formal errors, the file is automatically edited and may be corrected by the user by means of the usual CMS EDIT requests. Immediately afterwards, instead of giving other commands, the user may issue the GO command, which enables the program to proceed according to the expressions contained in the file. All new commands, written in at the terminal, overlay the commands already written in the file; more precisely, every new command is stored at the first record of the file, and is continued in the subsequent records only if, when one line is not sufficient, the command given contains the continuation sign (=).

This file is automatically allocated at the beginning of every work session, and automatically cancelled at the end. If it is used to store and execute a long series of commands, such as the complete set of regressions for estimating the coefficients of all the equations of a model (a quasi-batch work session), copy of the file under another name must be kept, in order to avoid its cancellation at the end of the work session. This capability can be regarded as a 'macro' function in order to use repeatedly any sequence of commands.

4. OPERATIONS AND FUNCTIONS

Issuing of commands and printing of results are carried out by the user through the terminal. The user must issue at the terminal the command to be executed, eventually followed by the names of the time series (from the central file or private series: in the event of the same names, the private series is the first to be searched for), numerical constants, operator codes and functions names.

Generally speaking, a command must be contained in one line, nevertheless the continuation on one or more subsequent lines is allowed, by using the special continuation sign (=) at the end of each line, except the last. The program decodes each line as soon as it is written, but it does not begin execution of the command until it comes to a line not ending with the continuation sign.

The numerical constants must be written with the decimal point, and they may be composed of up to 8 digits.

The names of the time series, up to 8 characters in length, must be written between quotes, in order to avoid confusion with the names of variables and the names of simple or composite operators.

The names of the functions can be written in free format; user defined functions can be up to 8 characters in length. In each case blank spaces may be added but they are ignored by the program.

The continuation sign must not split the names of variables, operator names, or numerical constants.

After executing command, the program prints the results on the terminal, but does not store these results. Should the user wish to retain these results (and this is only possible when the results constitute a time series or when a single numerical result is involved — thus excluding plotter output, for example), it is necessary to write the key word FILE at the end of the data string, following it up immediately — without any intermediate spaces — with the name of the file to be created: for example .... FILEGNP — the results of the command go into the FILEGNP P1, and may be used in all subsequent operations merely by referring to the GNP time series.

The following operations may be carried out on time series and on any constant:

1) Operations and functions executed by routines held in the storage (static routines).

.AND.	*Logical "and"*
.OR.	*Logical "or"*
.NOT.	*Logical "not"*
.LT.	*Less than*

.LE.	*Less or equal*
.EQ.	*Equal*
.NE.	*Not equal*
.GE.	*Greater or equal*
.GT.	*Greater than*
+	*Addition*
—	*Subtraction*
*	*Multiplication*
/	*Division*
**	*Raise to power*
LOG	*Natural logarithm*
LOG10	*Decimal logarithm*
EXP	*Exponential (base e)*
SIN	*Sine (Radians)*
COS	*Cosine (Radians)*
ATAN	*Arc tangent (Radians)*
SINH	*Hyperbolic sine*
COSH	*Hyperbolic cosine*
ABS	*Absolute value*
MAX	*Maximum value*
MIN	*Minimum value*
MEAN	*Mean value*
VAR	*Variance*
INT	*Integer part*
RV	*Rate of variation*
LAG (nn)	*Data lagging*
ONE (yy/mm)	*Selection of one single item of data from a series*
SEL (yy/mm,ll/kk)	*Selection of part of a series*
CPR (nn)	*Compression of a series*
IF	*For discontinuities and decision rules*

2) Operations and functions carried out by dynamic loading routines.

PLOT	*On line plotter*
CORR	*Correlation matrix*
OLS, OLSR	*Ordinary least squares estimate*
TSLS	*Two-stage least square estimate*
LISE	*Limited information single equation estimate*
DMS	*Connection to DMS/2*

4.1 Operator priority

Highest priority is given to the following functions: LOG, LOG10, EXP, SIN, COS, SINH, COSH, ATAN, ABS, MAX, MIN, MEAN, VAR, INT, RV, ONE, SEL, CPR, IF, OLSR, NOT. In the event of functions being chained, priority goes from right to left (the internal function being executed first).

These are followed, in order of precedence, by: raising to power, multiplication and division, in order from left to right, then addition and subtraction, in order from left to right.

Then comparison operations in order from left to right.

At the end, the logical operators .AND. and .OR. in order from left to right.

4.2 Arithmetic dyadic operators

+, −, *, /, **

The operator must be written between the two operands to which it refers. No special separator signs are necessary.

The execution of these operations is subject to the following rules. Between two constants, the result is a new constant. Between a constant and a series, the result is a new series with the same characteristics (starting date and finishing date, division into periods) as the original series. The operation between two series is allowed only if both series are divided into the same periods (for example, both with monthly data or both with annual data), in which case the result is a series limited to the period common to both the original series (for example, the sum of two series of annual data — one from 1953 to 1972 and the other from 1951 to 1970 — is a series of annual data from 1953 to 1970).

4.3 Logical dyadic operators

.AND., .OR., .LT., .LE., .EQ., .NE., .GE., .GT.

As for the arithmetic dyadic operators, the result may be a constant or a series, whose numerical values may be 0 (FALSE) or 1 (TRUE).

For the .AND. and .OR. operators, the operands can have any value, but they are tested only to control if the value is 0. (FALSE) or ≠0. (TRUE).

4.4 Monadic operators

1) EXP, LOG, LOG10, SIN, COS, ATAN, SINH, COSH, ABS, INT, .NOT.

The operator symbol must precede the variable (or constant) to which it is to be applied. It is possible to chain several operators, without using separator signs between them.

When applied to a constant, these functions transform it into a new constant. Applied to a series, they transform it into a new series with the same characteristics as the original one (starting date, finishing date, division into periods).

2) MAX, MIN, MEAN, VAR

They return only one numerical value, which has all the characteristics of a constant (there are no dates, etc.). However, if they are related to the operator CPR (compression of a series in conformity with another operator), they return a series (see CPR).

3) ONE (yy/mm)

Extracts from the series the numerical value corresponding to the year and the month (or quarter or half-year) indicated. For example, ONE(67/11)'SERIES1' returns the value for the month of November and for the year 1967 (SERIES1 must have monthly data); ONE(67)'SERIES1' returns the first (or only) item of data for the year 1967. The numerical value returned has all the characteristics of a constant.

4) LAG (nn)

This operator shifts the data of a time series (in one direction only) for as many positions as are indicated by nn (from 0 to 99). For example, LAG(4)'SERIES1', where SERIES1 (annual data) starts from 1961 and ends in 1969, returns a series starting from 1965 and ending in 1969, with the original value for 61 assigned to 65, etc.

5) SEL (yy/mm,ll/kk)

Selects from the series the values included between the year and the month (or quarter) indicated first, and that indicated second. For example SEL(55/10,66/2)'SERIES1' returns a series with monthly data (like SERIES1) included between October 1955 and February 1966.

6) RV

Calculates the variation rates of a series, taking into account the difference between one data and the preceding, whatever the periodicity. The results are in percentage form.

4.5 Composite operators

CPR (nn)

The function CPR (nn) (compression of a series according to another operator) must be used together with one of the operators MAX, MIN, MEAN, VAR, +, *,: it must never be used alone. This function gives a series with characteristics as illustrated in the following examples: MAX CPR(1)'SERIES1', (where SERIES1 has monthly data) gives a series of annual values (1 per year) equal to the maximum value among the monthly values for each year; MEAN CPR(4)'SERIES1' gives a series of quarterly values (4 per year) equal to the mean from among the values for each quarter in the series; 0. + CPR(2)'SERIES1' gives a series of half-yearly values (2 per year) equal to the sum of all the values for each half year in the series indicated. In these cases, the operators MAX, MIN, MEAN, VAR give a series instead of a constant.

4.6 The IF operator

Discontinuities and decision rules can be analyzed by means of the IF operator. It is used together with the pseudo-operators THEN and ELSE as illustrated in the following examples:

1) IF('SERIES1' .LT. 'SERIES2') THEN'SERIES3' ELSE'SERIES4'

First of all the program performs the operation with highest priority (.LT. because it is enclosed in parenthesis); the result is a series, limited to the period common to SERIES1 and SERIES2 and with the same division into periods, with numerical values 0. and 1.

This resulting series should have the same division into periods as SERIES3 and SERIES4 in order to perform the operation for the period common to these three series, selecting for every year the value from SERIES3 if the corresponding value of the logical calculated series is nonzero (TRUE) or from SERIES4 if it is 0. (FALSE). These values build the resulting series.

2) IF('SERIES1' .LT. 1000.) THEN'SERIES2' ELSE IF(178..LT.'SERIES3') THEN'SERIES4' ELSE 250.

In this example, comparisons are made between series and constants (every value of the series with the constant) and the resulting values are selected from the series or from the constant. Each ELSE refers to the last unclosed THEN; each THEN must be closed by an ELSE.

4.7 Special functions

This group includes all dynamic-loading routines in the vacant high-storage address area, that is to say on-line plotter routines, estimate routines (OLS, OLSR, TSLS, LISE), the routine for connection to DMS/2, the routine for the correlation matrix and user defined functions.

1) OLS (ordinary least squares)

Multiple linear regression by the ordinary least squares method is executed by means of the SSP (Scientific Subroutine Package) library routines [4]. On the basis of the ordinary least squares method, it is possible with these routines to calculate the standard deviations and means of both dependent and independent variables, simple and multiple correlation coefficients, and also to calculate regression coefficients.

Variance analysis is also executed and, on request, the residue table (differences between the observed and interpolated values) is printed. On request, it is possible to specify the output unit (6 = tele-type terminal, 8 = high-speed printer) when printed output of the results is required. For example OLS('SERIES1', RV('SERIES2' + 'SERIES3'), 'SERIES4'); SERIES1 is the dependent variable (written first in the list of variables separated by commas), while the other two, that is, the series resulting from the RV('SERIES2' + 'SERIES3') operation and 'SERIES4' are the independent variables.

2) OLSR

It performs the same operation of OLS, but instead of printing a table of statistics, it gives, as a result, the series of the residuals of the linear regression. It can be so used as each other function. For example: 'SERIES1' – OLSR('SERIES1', 'SERIES2'); the result is the series of the interpolated values of SERIES1.

PLOT('SERIES2', OLSR('SERIES1', 'SERIES2')) (see PLOT) plots on the terminal the residuals against the explaining variable SERIES2.

Among the various single-equation methods for estimating systems of simultaneous equations, TSLS and LISE are considered hereunder.

3) TSLS (two-stage least squares)

The command TSLS ('SERIES1', 'SERIES2', ...) sorts all the variables involved in the system to be estimated: all the endogenous variables are defined first, then the predetermined variables. After preliminary control of the total number of variables sorted, by means of specification of the number of endogenous and predetermined variables, the program executes the OLS estimate of the 'reduced form' (first stage).

Then (second stage), for each structural equation (at most as many as the endogenous variables), the user must insert the specifications (number of variables and sorting index) needed for sorting the subset of variables involved in the equation. For each equation, the estimate of the coefficients and the estimate of the standard error referred to each coefficient are supplied.

4) LISE (limited information single equation)

The operational characteristics of the command LISE are the same specified for TSLS; however, even if the user is not aware of this, it should be noted that the 'reduced form' estimate applies only to the 'standard error' estimate, while for estimating the coefficients of structural equations, in the iterative process which leads to the maximization of the variance ratio, the OLS estimate of the same equation is taken as the point of departure.

5) PLOT

The PLOT command permits the printing on the terminal of a graph in points for certain (dependent) variables versus another (independent) variable. The scale is automatically adjusted on the basis of the minimum and maximum values of the variables. Example: PLOT('SERIES1', 'SERIES2', RV'SERIES3' * MAX'SERIES3'/100.); 'SERIES1' is the independent variable (abscissa), while 'SERIES2' and the series resulting from the RV'SERIES3' * MAX'SERIES3' /100 operation are the dependent variables (ordinates).

In conversational mode, the user is asked to choose between two different types of on-line plotter, that is — the normal plotter, with numerical values and scales, which takes considerable time to print, or the high-speed, qualitative plotter, without numerical values and scales which, especially in the case of a single dependent variable, permits a considerable saving in time.

6) DMS

The command DMS ('SERIES1', 'SERIES2' ......) allows the selection of a certain number of time series, writing their names and numerical values on the FILE DMS P1, according to the format required by the DMS/2 input; the data are, however, arranged variable by variable, while DMS/2 requires them arranged year by year. It is therefore necessary to use the CMS SORT command which permits rearrangement of the file in date order. In this case, series of different lengths are not cut, as this operation is handled by DMS/2.

7) CORR

The command CORR ('SERIES1','SERIES2'.....) allows to obtain the symmetric correlation matrix of the selected variables. Only the lower part of the matrix is printed out.

4.8 User defined functions

The user himself can create special functions of interest for him, according to the general philosophy of the IMTS. He must write a program whose FILENAME must be the name he wants to give the function; the name should be different from those used by the IMTS operations and functions.

Any program (if written in FORTRAN language) must begin with the following statement: SUBROUTINE USER (A, B, C, D, ...) where A, B, C, D, is a standard list of arguments. An example will explain the creation and usage of a function.

The user wants to create the function UPTOMAX that, when applied to a time series, returns a new series whose values is, for every year, the maximum value up to that year.

Using the CMS editing facilities, he creates the program UPTOMAX FORTRAN P1, in which he 'GET's (see editing for CMS) the program USER FORTRAN A1, that contains only the statement SUBROUTINE USER (A, B, C, D, ...) and the comments explaining the meaning of the arguments. Following that first set of statements, the program deals with the desired function, taking into account that the subroutine parameters will be automatically transferred when the function is used.

After a correct compilation, this function is immediately available with no other formalities and can be used in the same way as the IMST commands.

When the decoding routine, during the scanning of the command, finds the unknown symbol UPTOMAX, instead of returning an error code looks in the user's P-disk for a file whose filetype is TEXT and filename is UPTOMAX' (8 characters); as it does not find, it looks for a filename one character shorter (7) that is UPTOMAX; as it finds (if not, it would have tried with 6,5,4,3,2,1 characters), the IMTS loads into the high address free area a standard program named INTRFACE and the UPTOMAX subroutine (INTRFACE is the main program that contains the statement CALL USER (A, B, C, D, ...)), starts INTRFACE, passing to it all the necessary parameters (values of the series, number of selected series, lenght of each series, starting and finishing date and so on) and takes them back as a resulting series, that can be processed by other functions in the same command (added to SERIES1 and SERIES2 in the above example).

User defined functions are treated as the other functions for the priority.

4.9 Results Format

Leaving aside the special plotter and regression functions, which have special output characteristics, the results of processing work of any other kind are subject to the following format standards.

The FILEaaaaaaa option written at the end of a command makes it possible to preserve the results, storing them in FILE aaaaaa (any name) P1, which may be used as input in a subsequent command.

If this option is omitted, the results are printed at the terminal and cannot be used again.

If the characteristics of the final results are those of a constant, only the numerical datum is printed (or written into a work file); but if the characteristics are those of a series, the numerical data are printed (or placed in a file) with indication of the corresponding years and months (or half-years, etc.).

In the case of time series printed at the terminal, there are two printing methods available: the first of these methods is called into operation by printing between quotes at the terminal only the name of one single time series contained in the central file or in one of the other files, without the addition of any operators; in this case, as well as the numerical data, the other data are also printed (full name, unit of measurement, etc.): for example: 'SERIES1'. The second method is used in all other cases, even when a dummy operation (for example — 'SERIES1' + 0.) is performed on one single series; in this case only the numerical results are printed.

In both cases the numerical data are printed one per line, with 6 meaningful digits and, alongside each item of data, indication of the year and month (or half-year or quarter) to which they refer.

In the case of time series to be stored in a file, the usual format is used (as described in 3-3), but the full name and the unit of measurement and data source are left blank because, after processing, they may lose their significance.

4.10 Formal error recovery procedure

During the operations for the coding and decoding of commands, some types of formal errors are likely to occur in the program (for a complete list see n. 7) as, for example, number of open parenthesis different from number of closed parenthesis, quotes opened, but not closed, to contain names of series, names of series exceeding 8 characters, and so on.

After reporting the type of error and the name of the routine containing the error (indication normally not essential for the user, but very useful for the system engineer in setting up new functions), the program calls the EDIT program of the CMS and applies it to the GO IMTS file, that is to say to the file in which each command is stored before decoding.

Using the usual EDIT requests, the user corrects the given command, cancels from the file any other records not required, gives the FILE command, which retains the file content and returns control to the main program, then, if required, gives the GO command to repeat operation of the corrected command. If this still contains formal errors, the procedure starts again. If the user prefers to rewrite the stored command entirely instead of correcting it, he only needs to leave EDIT environment by means of the QUIT command, which leaves the file as it is, and does not give the GO command.

Other types of errors are also contemplated as, for example, time series which are not present either in the central file or in the user's P-DISK, operations which do not yield any result, for example arithmetic operations on two incompatible series, one with monthly data and one with annual data, or on two compatible series without a common period, divisions by zero, logarithms of negative numbers, and so on.

In the case of all these errors, which are not in the formal error category, the correction procedure is not called automatically.

Should he think it advisable, however, the user may edit the GO IMTS file and then proceed as for formal errors.

5. UTILITY PROGRAMS

The category of utility programs includes all routines in the operating system (held in the CMS core, transient routines or those held in the disk system) accessible to the user (EDIT, COMBINE, LISTF, SORT, APL, etc.) and certain special programs which are external to the main program but are accessible from within it (CREARCH, NEW). The question of storage allocation for these programs is dealt with in section 2.

The CMS routines are accessed from within the program by means of the usual CMS commands. As the special programs are held on the same disk as the central file in module format, they are called by typing only their name.

1) CREARCH

This program was used at the outset to create the central file, and it is used whenever new time series are to be added to this file. It may be used only by the virtual machine which handles the file; the file, according to the program philosophy, may be read contemporaneously by more than one virtual machine, but may be modified by only one.

2) NEW

This is a program which permits the creation of a file containing a new time series in user's P-DISK, without any problem. It reads the data as and when they are written and then stores them in the required file, in correct format. Once it has been created and completed, the file may be held in the P-DISK or sent to the virtual machine, which has read/write access to the central file.

In the later case the series become available for all the users.

6. DISKS ORGANIZATION

The file, the data processing program and the utility programs are contained in one single CMS minidisk. Only one virtual machine has read/write access to this disk. This means that the file may be updated only by the same virtual machine.

All the other authorized virtual machines have READ—ONLY access to this disk by means of the following CMS command: LOGIN 193 A,P; this command may be given automatically by means of an EXEC procedure.

The work files for each virtual machine are created in the work disk of the respective machine, and may therefore be manipulated by the user.

The data processing program is in MODULE format; it may therefore be called merely by writing the IMTS command at the terminal.

As soon as the program takes over control, an arrow is printed at the terminal and the keyboard is released, pending indication of the operations to be carried out. On completion of processing, the program restarts from the beginning, repeating the arrow.

The IMTS command must be issued in the CMS environment; to close it and reenter the CMS, it is sufficient to write FINE in columns 1-4.

7. ERROR MESSAGES

7.1 Formal errors

For the following errors an automatic error recovery procedure is supplied (see 4-10):

(D1) Unexpected character in non-blank card column XX

(D1) 's do not balance or name longer than 8 characters

(D1) First argument of operand LAG, or ONE, or SEL, or CPR not allowed

(D2) The name whose first 8 characters are XXXXXXXX is longer than the allowed format

(D2) The operand is not acceptable

(D2) The expression is not allowed

(D2) Parenthesis or THEN-ELSE do not balance

7.2 Non-formal errors

For the following errors the error recovery procedure is not automatically called (see 4-10):

(D3) Incorrect number of data in FILE XXXXXXXX

(D3) FILE FT01F001 cannot be used as input file

(T5) XXXXXXXX not-existent series

(T6) XXXXXXXX: this series has no continuation

(D4) The divisor in zero in row XXXXXXXX

(D4) The base XXX is not positive while the exponent XXX not integer (row XXXXXXXX)

(D4) Logarithm of a XXXXXXXX not positive number (row XXXXXXXX)

(D4) The series XXXXXXXX and YYYYYYYY cannot be compared due to their different division into periods

(D4) Insufficient number of data to perform the requested LAG, RV or DIF operation on series XXXXXXXX

(D4) The division into periods of the series XXXXXXXX is incompatible with the requested CPR operation

(D4) No result

(R1) No result

(R1) Insufficient number of series

(R1) Different division into periods of the series involved in the IF operation

(OLS) The matrix is singular

(OLS) Number of data < = number of variables

(PLOT) Maximum and minimum values of the dependent or independent variables are equal: plotting is impossible.

8. EXAMPLE OF A WORK SESSION AT THE 2741 TERMINAL

```
CP ENTERED, REQUEST, PLEASE.
CP
i cms
CMS..VERSION  3.0 (7 nov 73)


R; T=0.03/0.11 16.09.43

login 193 a,p
** A (193) READ-ONLY **
R; T=0.03/0.11 16.09.58

imts

--->
```

For the following example see KMENTA [5, pag. 563]

```
'qt'

NUMERO DELLA SERIE:        0 number of the time series
NOME ABBREVIATO:           QT abbreviated name
ANNO INIZIALE:             1922  first year
ANNO FINALE:               1941  last year
NOME INTERO:               FOOD CONSUMPTION PER HEAD USA  complete name
FONTE DEI DATI:            KMENTA PAG.565  source of data
UNITA' DI MISURA:           unit of measurements:
I DATI SONO ANNUALI.        annual data
NUMERI INDICI.              index number

22/ 1       98.4850
23/ 1       99.1870
24/ 1       102.163
25/ 1       101.504
26/ 1       104.240
27/ 1       103.243
28/ 1       103.993
29/ 1       99.9000
30/ 1       100.350
31/ 1       102.820
32/ 1       95.4350
33/ 1       92.4240
34/ 1       94.5350
35/ 1       98.7570
36/ 1       105.797
37/ 1       100.225
38/ 1       103.522
39/ 1       99.9290
40/ 1       105.223
41/ 1       106.232

--->
```

```
--->
'pt'

NUMERO DELLA SERIE:      0
NOME ABBREVIATO:         PT
ANNO INIZIALE:           1922
ANNO FINALE:             1941
NOME INTERO:             RATIO OF FOOD PRICES TO GENERAL CONSUM.PRICES USA
FONTE DEI DATI:          KMENTA PAG.565
UNITA' DI MISURA:
I DATI SONO ANNUALI.
NUMERI INDICI.

22/ 1     100.323
23/ 1     104.264
24/ 1     103.435
25/ 1     104.506
26/ 1     98.0010
27/ 1     99.4560
28/ 1     101.066
29/ 1     104.763
30/ 1     96.4460
31/ 1     91.2280
32/ 1     93.0850
33/ 1     98.8010
34/ 1     102.908
35/ 1     98.7560
36/ 1     95.1190
37/ 1     98.4510
38/ 1     86.4980
39/ 1     104.016
40/ 1     105.769
41/ 1     113.480

--->
```

```
---
'dt'

NUMERO DELLA SERIE:      0
NOME ABBREVIATO:         DT
ANNO INIZIALE:           1922
ANNO FINALE:             1941
NOME INTERO:             DISPOSABLE INCOME IN CONSTANT PRICES    USA
FONTE DEI DATI:          KMENTA PAG.565
UNITA' DI MISURA:
I DATI SONO ANNUALI.
NUMERI INDICI.

22/ 1      87.4000
23/ 1      97.6000
24/ 1      96.7000
25/ 1      98.2000
26/ 1      99.8000
27/ 1      100.500
28/ 1      103.200
29/ 1      107.800
30/ 1      96.6000
31/ 1      88.9000
32/ 1      75.1000
33/ 1      76.9000
34/ 1      84.6000
35/ 1      90.6000
36/ 1      103.100
37/ 1      105.100
38/ 1      96.4000
39/ 1      104.400
40/ 1      110.700
41/ 1      127.100

--->
```

```
--->
'ft'

NUMERO DELLA SERIE:     0
NOME ABBREVIATO:        FT
ANNO INIZIALE:          1922
ANNO FINALE:            1941
NOME INTERO:            PRICES RECEIVED BY FARMERS USA
FONTE DEI DATI:         KMENTA PAG.565
UNITA' DI MISURA:
I DATI SONO ANNUALI.
NUMERI INDICI.

22/ 1      98.0000
23/ 1      99.1000
24/ 1      99.1000
25/ 1      98.1000
26/ 1      110.800
27/ 1      108.200
28/ 1      105.600
29/ 1      109.800
30/ 1      108.700
31/ 1      100.600
32/ 1      81.0000
33/ 1      68.6000
34/ 1      70.9000
35/ 1      81.4000
36/ 1      102.300
37/ 1      105.000
38/ 1      110.500
39/ 1      92.5000
40/ 1      89.3000
41/ 1      93.0000

--->
```

```
--->
'time'

NUMERO DELLA SERIE:     0
NOME ABBREVIATO:        TIME
ANNO INIZIALE:          1922
ANNO FINALE:            1941
NOME INTERO:            TIME IN YEARS  (1922=1)
FONTE DEI DATI:         .....
UNITA' DI MISURA:       ....
I DATI SONO ANNUALI.
LIVELLI.

22/ 1      1.00000
23/ 1      2.00000
24/ 1      3.00000
25/ 1      4.00000
26/ 1      5.00000
27/ 1      6.00000
28/ 1      7.00000
29/ 1      8.00000
30/ 1      9.00000
31/ 1      10.0000
32/ 1      11.0000
33/ 1      12.0000
34/ 1      13.0000
35/ 1      14.0000
36/ 1      15.0000
37/ 1      16.0000
38/ 1      17.0000
39/ 1      18.0000
40/ 1      19.0000
41/ 1      20.0000

--->
```

```
--->
'cost'

NUMERO DELLA SERIE:      0
NOME ABBREVIATO:         COST
ANNO INIZIALE:           1922
ANNO FINALE:             1941
NOME INTERO:             CONSTANT TERM FOR THE INTERCEPT
FONTE DEI DATI:          .....
UNITA' DI MISURA:        .....
I DATI SONO ANNUALI.
LIVELLI.

22/ 1      1.00000
23/ 1      1.00000
24/ 1      1.00000
25/ 1      1.00000
26/ 1      1.00000
27/ 1      1.00000
28/ 1      1.00000
29/ 1      1.00000
30/ 1      1.00000
31/ 1      1.00000
32/ 1      1.00000
33/ 1      1.00000
34/ 1      1.00000
35/ 1      1.00000
36/ 1      1.00000
37/ 1      1.00000
38/ 1      1.00000
39/ 1      1.00000
40/ 1      1.00000
41/ 1      1.00000

--->
```

The above listed series are private series, contained in the user's P-DISK.

```
--->
tsls ('qt','pt','dt','ft','time','cost')

THERE ARE  20 DATA FOR EACH SERIES, FROM 1922/ 1 TO 1941/ 1 :  1 EACH YEAR.
EXECUTION BEGINS...
 ENTER NUMBERS OF SELECTED ENDOGENOUS AND PREDETERMINED VARIABLES
--..
0204


END.      =INDEX OF EXPLAINED ENDOGENOUS VARIABLE  (00=EXIT)
N.ENEX    =NUMBER OF ENDOGENOUS EXPLANATORY VARIABLES
N.EXEX    =NUMBER OF EXOGENOUS  EXPLANATORY VARIABLES
IND.ENEX =INDEX END. EXPLAN.
IND.EXEX =INDEX EXO. EXPLAN.


END.  N.ENEX  N.EXEX
..--..
010102
IND. ENEX
..--..--..--..--
02
IND. EXEX
..--..--..--..--
0306


TWO-STAGE LEAST SQUARES ESTIMATION

DIP. VAR.      Y 1

IND.VAR.       EST. COEFF.    EST. STD.ERROR
  Y 2           -0.24356      0.09648
  X 3            0.31399      0.04694
  X 6           94.63327      7.92084

STD. ERROR OF ESTIMATE      1.96632
DURBIN-WATSON STATISTIC     2.00922
```

```
END.  N.ENEX  N.EXEX
..--..
010103
IND. ENEX
..--..--..--..--
02
IND. EXEX
..--..--..--..--
040506


TWO-STAGE LEAST SQUARES ESTIMATION

DIP. VAR.      Y 1

IND.VAR.       EST. COEFF.    EST. STD.ERROR
  Y 2            0.24008       0.09993
  X 4            0.25561       0.04725
  X 5            0.25292       0.09966
  X 6           49.53241      12.01053

STD. ERROR OF ESTIMATE         2.45756
DURBIN-WATSON STATISTIC        2.38465

--->
```

```
--->
llse('qt','pt','dt','ft','time','cost')

THERE ARE  20 DATA FOR EACH SERIES, FROM 1922/ 1 TO 1941/ 1 :  1 EACH YEAR.
EXECUTION BEGINS...
 ENTER NUMBERS OF SELECTED ENDOGENOUS AND PREDETERMINED VARIABLES
--..
0204


END.      =INDEX OF EXPLAINED ENDOGENOUS VARIABLE  (00=EXIT)
N.ENEX    =NUMBER OF ENDOGENOUS EXPLANATORY VARIABLES
N.EXEX    =NUMBER OF EXOGENOUS  EXPLANATORY VARIABLES
IND.ENEX =INDEX END. EXPLAN.
IND.EXEX =INDEX EXO. EXPLAN.


END.  N.ENEX  N.EXEX
..--..
010102
IND. ENEX
..--..--..--..--
02
IND. EXEX
..--..--..--..--
0306


LIMITED INFORMATION SINGLE EQUATION

DIP. VAR.    Y 1

IND.VAR.      EST. COEFF.    EST. STD.ERROR
  Y 2          -0.22954      0.09800
  X 3           0.31001      0.04743
  X 6          93.61922      8.03125

STD. ERROR OF ESTIMATE       1.98142
DURBIN-WATSON STATISTIC      2.05172
```

```
END.  N.ENEX  N.EXEX
..--..
010103
IND. ENEX
..--..--..--..--
02
IND. EXEX
..--..--..--..--
040506


LIMITED INFORMATION SINGLE EQUATION

DIP. VAR.     Y 1

IND.VAR.       EST. COEFF.    EST. STD.ERROR
  Y 2            0.24008      0.09993
  X 4            0.25561      0.04725
  X 5            0.25292      0.09966
  X 6           49.53241     12.01053

STD. ERROR OF ESTIMATE        2.45756
DURBIN-WATSON STATISTIC       2.38465

--->
```

```
--->
ols('qt','pt','dt')

THERE ARE  20 DATA FOR EACH SERIES, FROM 1922/ 1 TO 1941/ 1 :  1 EACH YEAR.
EXECUTION BEGINS...
1 SIMBOLIC UNIT (6/8)
2-9 NAME
10 RESIDUALS (0/1)
-........-
6demand  0
MULTIPLE REGRESSION.....DEMAND


VARIABLE        CORRELATION     REGRESSION     STD. ERROR      COMPUTED
  NO.             X VS Y        COEFFICIENT    OF REG.COEF.    T VALUE
   2             0.09810        -0.31630        0.09068        -3.48817
   3             0.77118         0.33464        0.04542         7.36728
DEPENDENT
   1

 INTERCEPT                      99.89539
MULTIPLE CORRELATION            0.87395
R**2                            0.76379
STD. ERROR OF ESTIMATE          1.93013
STD. DETERMINANT                0.67902

                    A N O V A
  SOURCE     D.F.      SUM OF        MEAN        F VALUE
                       SQUARES       SQUARES
REGRESSION    2       204.7826      102.3913    27.48468
RESIDUAL     17        63.3317        3.7254
   TOTAL     19       268.1144

            DURBIN WATSON STATISTIC = 1.7442


--->
```

```
--->
ols ('qt','pt','ft','time')

THERE ARE  20 DATA FOR EACH SERIES, FROM 1922/ 1 TO 1941/ 1 :  1 EACH YEAR.

1 SIMBOLIC UNIT (6/8)
2-9 NAME
10 RESIDUALS (0/1)
-........-
6supply  0
MULTIPLE REGRESSION.....SUPPLY


VARIABLE        CORRELATION     REGRESSION      STD. ERROR      COMPUTED
  NO.             X VS Y        COEFFICIENT     OF REG.COEF.    T VALUE
   2               0.09810        0.16037         0.09488        1.69014
   3               0.68057        0.24813         0.04619        5.37226
   4               0.15004        0.24830         0.09752        2.54622
DEPENDENT
   1

 INTERCEPT                        58.27541
MULTIPLE CORRELATION              0.80920
R**2                              0.65481
STD. ERROR OF ESTIMATE            2.40509
STD. DETERMINANT                  0.88357

                      A N O V A
  SOURCE      D.F.        SUM OF        MEAN        F VALUE
                          SQUARES       SQUARES
REGRESSION     3         175.5632       58.5211     10.11697
RESIDUAL      16          92.5512        5.7844
   TOTAL      19         268.1144

            DURBIN WATSON STATISTIC =  2.1097




--->
```

```
--->
go
OLS ('QT','PT','FT','TIME')


THERE ARE  20 DATA FOR EACH SERIES, FROM 1922/ 1 TO 1941/ 1 :  1 EACH YEAR.
1 SIMBOLIC UNIT (6/8)
2-9 NAME
10 RESIDUALS (0/1)
-........-
6supply  1
MULTIPLE REGRESSION.....SUPPLY
```

VARIABLE NO.	CORRELATION X VS Y	REGRESSION COEFFICIENT	STD. ERROR OF REG.COEF.	COMPUTED T VALUE
2	0.09810	0.16037	0.09488	1.69014
3	0.68057	0.24813	0.04619	5.37226
4	0.15004	0.24830	0.09752	2.54622
DEPENDENT				
1				

INTERCEPT	58.27541
MULTIPLE CORRELATION	0.80920
R**2	0.65481
STD. ERROR OF ESTIMATE	2.40509
STD. DETERMINANT	0.88357

A N O V A

SOURCE	D.F.	SUM OF SQUARES	MEAN SQUARES	F VALUE
REGRESSION	3	175.5632	58.5211	10.11697
RESIDUAL	16	92.5512	5.7844	
TOTAL	19	268.1144		

TABLE OF RESIDUALS

CASE NO.	Y VALUE	Y ESTIMATE	RESIDUAL
1	98.48499	98.92925	-0.44427
2	99.18700	100.08250	-0.89551
3	102.16299	100.19786	1.96513
4	101.50400	100.36978	1.13421
5	104.23999	102.72619	1.51380
6	103.24300	102.56268	0.68031
7	103.99300	102.42403	1.56897
8	99.89999	104.30736	-4.40737
9	100.34999	102.94895	-2.59896
10	102.81999	100.35058	2.46941
11	95.43500	96.03327	-0.59827
12	92.42400	94.12138	-1.69738
13	94.53499	95.59901	-1.06402
14	98.75699	97.78687	0.97012
15	105.79700	102.63790	3.15910
16	100.22499	104.09051	-3.86552
17	103.52199	103.78668	-0.26469
18	99.92899	102.37789	-2.44890
19	105.22299	102.11328	3.10971
20	106.23199	104.51787	1.71412

DURBIN WATSON STATISTIC = 2.1097

```
--->
```

For the following series, contained in the central file, see the ISCO model [6, pag. 118]

```
--->
'di'

NUMERO DELLA SERIE:      4
NOME ABBREVIATO:         DI
ANNO INIZIALE:           1951
ANNO FINALE:             1970
NOME INTERO:             DOMANDA INTERNA GLOBALE
FONTE DEI DATI:          (CPR+CPA+I)
UNITA' DI MISURA:        MILIARDI LIRE 1963
I DATI SONO ANNUALI.
LIVELLI.

51/ 1      15647.0
52/ 1      16788.0
53/ 1      17902.0
54/ 1      18552.0
55/ 1      19542.0
56/ 1      20498.0
57/ 1      21467.0
58/ 1      22311.0
59/ 1      23569.0
60/ 1      25248.0
61/ 1      27218.0
62/ 1      29159.0
63/ 1      31530.0
64/ 1      31801.0
65/ 1      31943.0
66/ 1      33773.0
67/ 1      36359.0
68/ 1      38444.0
69/ 1      40844.0
70/ 1      43486.0

--->
```

```
--->
'k'

NUMERO DELLA SERIE:    27
NOME ABBREVIATO:        K
ANNO INIZIALE:          1951
ANNO FINALE:            1972
NOME INTERO:            CAPACITA' OCCUPATA NEL SETTORE INDUSTRIALE
FONTE DEI DATI:         ISCO FUA DOCUMENTO 14 '72 PG.111
UNITA' DI MISURA:       QUOTA PERCENTUALE
I DATI SONO ANNUALI.
LIVELLI.

51/ 1      93.1000
52/ 1      88.7000
53/ 1      88.9000
54/ 1      90.2000
55/ 1      90.9000
56/ 1      90.4000
57/ 1      90.2000
58/ 1      86.6000
59/ 1      88.7000
60/ 1      92.7000
61/ 1      94.5000
62/ 1      94.9000
63/ 1      95.8000
64/ 1      89.1000
65/ 1      85.3000
66/ 1      88.3000
67/ 1      90.4000
68/ 1      91.3000
69/ 1      89.6000
70/ 1      90.6000
71/ 1      86.0000
72/ 1      85.7000

--->
```

```
--->
'mm'

NUMERO DELLA SERIE:    35
NOME ABBREVIATO:        MM
ANNO INIZIALE:          1952
ANNO FINALE:            1972
NOME INTERO:            IMPORTAZIONI DI MERCI
FONTE DEI DATI:         ISTAT IMP.CIF ANN.STAT.COMM.EST.
UNITA' DI MISURA:       MLRD 63 DEFL=PMM
I DATI SONO ANNUALI.
LIVELLI.

52/ 1      1145.00
53/ 1      1287.00
54/ 1      1352.00
55/ 1      1482.00
56/ 1      1681.00
57/ 1      1848.00
58/ 1      1838.00
59/ 1      2070.00
60/ 1      2941.00
61/ 1      3327.00
62/ 1      3863.00
63/ 1      4744.00
64/ 1      4379.00
65/ 1      4434.00
66/ 1      5097.00
67/ 1      5777.00
68/ 1      6019.00
69/ 1      7194.00
70/ 1      8323.00
71/ 1      8176.00
72/ 1      9155.00

--->
```

If the command contains a formal error:

```
--->
ols(log'mm',log'dl',log'k'
(D2)ERROR : PARENTHESIS OR THEN-ELSE DO NOT BALANCE
(AMDM1) ERROR RECOVERY PROCEDURE
USE EDIT COMMANDS.
DEFAULT TABS SET.
EDIT:
p
OLS(LOG'MM',LOG'DI',LOG'K'
c /k' /k')/
OLS(LOG'MM',LOG'DI',LOG'K')
file

--->
go
OLS(LOG'MM',LOG'DI',LOG'K')


THERE ARE  19 DATA FOR EACH SERIES, FROM 1952/ 1 TO 1970/ 1 :  1 EACH YEAR.
1 SIMBOLIC UNIT (6/8)
2-9 NAME
10 RESIDUALS (0/1)
-........-
6import  0
MULTIPLE REGRESSION.....IMPORT


VARIABLE       CORRELATION    REGRESSION     STD. ERROR      COMPUTED
  NO.            X VS Y       COEFFICIENT    OF REG.COEF.    T VALUE
   2            0.99567        2.12933        0.04094       52.01243
   3            0.18790        1.19420        0.41229        2.89648
DEPENDENT
   1

 INTERCEPT                  -19.07669
MULTIPLE CORRELATION          0.99716
R**2                          0.99433
STD. ERROR OF ESTIMATE        0.05089
STD. DETERMINANT              0.98193

                 A N O V A
  SOURCE     D.F.      SUM OF        MEAN        F VALUE
                       SQUARES      SQUARES
REGRESSION    2         7.2646       3.6323 1402.44488
RESIDUAL     16         0.0414       0.0026
   TOTAL     18         7.3061

          DURBIN WATSON STATISTIC =  1.3079



--->
```

```
--->
ols (sel(55,68)log'mm',log'dl',log'k')

THERE ARE  14 DATA FOR EACH SERIES, FROM 1955/ 1 TO 1968/ 1 :  1 EACH YEAR.
1 SIMBOLIC UNIT (6/8)
2-9 NAME
10 RESIDUALS (0/1)
-........-
6import  0
MULTIPLE REGRESSION.....IMPORT


VARIABLE        CORRELATION    REGRESSION     STD. ERROR      COMPUTED
  NO.             X VS Y       COEFFICIENT    OF REG.COEF.    T VALUE
   2             0.99098         2.19860        0.06996       31.42485
   3             0.15116         1.24743        0.46026        2.71026
DEPENDENT
   1

 INTERCEPT                       -20.02585
MULTIPLE CORRELATION               0.99460
R**2                               0.98924
STD. ERROR OF ESTIMATE             0.05573
STD. DETERMINANT                   0.99549

                      A N O V A
  SOURCE      D.F.       SUM OF        MEAN       F VALUE
                        SQUARES       SQUARES
REGRESSION     2          3.1395        1.5698   505.43478
RESIDUAL      11          0.0342        0.0031
   TOTAL      13          3.1737

           DURBIN WATSON STATISTIC =  1.3591




--->
fine
CMS..VERSION  3.0 (7 nov 73)
```

9. BIBLIOGRAPHY

1) IBM — (CP-67/CMS) Version 3
CMS Program Logic Manual - GY20-0591-0

2) IBM — (CP-67/CMS) Version 3
User's Guide - GH20-0859-0

3) P. CORSI, A. STAJANO — An interactive programming system to solve econometric non linear models
Technical Report - CSP010/513-3505, IBM Italia

4) IBM — System/360 Scientific Subroutine Package Version III
Programmer's manual - H20-0203

5) J. KMENTA — Elements of Econometrics, The Macmillan Company
New York, 1971

6) M. CRIVELLINI — Analisi quantitativa per la programmazione di breve periodo (Ricerca di gruppo coordinata da G. Fuá)
Settore reale: struttura e simulazione
Documento di lavoro n. 15, March 1972